Engineering Applications of Artificial Intelligence 26 (2013) 1230–1236

Contents lists available at SciVerse ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

A real-time dynamic optimal guidance scheme using a general regression neural network

M.A. Hossain [a], A.A.M. Madkour [a], K.P. Dahal [b], Li Zhang [a,*]

[a] Computational Intelligence Group, School of Computing, Engineering and Information Sciences, Northumbria University, Newcastle, UK
[b] Artificial Intelligence Research Group, School of Computing and Media, University of Bradford, Bradford, UK

ARTICLE INFO

Article history:
Received 25 April 2012
Received in revised form
8 August 2012
Accepted 9 October 2012
Available online 31 October 2012

Keywords:
Optimal guidance algorithms
Proportional navigation guidance
Genetic algorithm
General regression neural network
Computational complexity
Real-time solution

ABSTRACT

This paper presents an investigation into the challenges in implementing a hard real-time optimal non-stationary system using general regression neural network (GRNN). This includes investigation into the dynamics of the problem domain, discretisation of the problem domain to reduce the computational complexity, parameters selection of the optimization algorithm, convergence guarantee for real-time solution and off-line optimization for real-time solution. In order to demonstrate these challenges, this investigation considers a real-time optimal missile guidance algorithm using GRNN to achieve an accurate interception of the maneuvering targets in three-dimension. Evolutionary Genetic Algorithms (GAs) are used to generate optimal guidance training data set for a large missile defense space to train the GRNN. The Navigation Constant of the Proportional Navigation Guidance and the target position at launching are considered for optimization using GAs. This is achieved by minimizing the miss distance and missile flight time. Finally, the merits of the proposed schemes for real-time accurate interception are presented and discussed through a set of experiments.

1. Introduction

Despite high performance computing power, real-time implementation of a complex evolutionary algorithm requires an efficient design, software coding of the algorithm, algorithm manipulation so as to exploit special features of the hardware and avoid associated shortcomings of the architecture. There have been many effort made earlier to demonstrate efficient algorithm design and software coding to achieve real-time performance (Benoit et al., 2010). However, computationally complex real-time applications, for example optimal algorithms to solve nonstationary system design problem in a dynamic environment, are difficult to implement by exploiting the power of high performance computing domain and/or efficient algorithm design technique and/or better software coding. In particular, the requirement of any level of online optimization for large number of data brings hard constraints and uncertainty in implementing the application in real-time. An alternative solution is to exploit algorithm manipulation, for example, performing off-line optimsation and then replacing database of optimal parameters with learning algorithms for online execution. In implementing a computationally complex scenario like this, an artificial neural network algorithm could play a vital role in learning the dynamic features in order to achieve the real-time solution of complex evolutionary problems.

This paper considers a closed loop complex real-time missile guidance scheme to demonstrate the potentiality of the GRNN algorithm to solve an evolutionary nonstationary problem. This requires optimal dynamic inputs to the scheme based on the nonstationary enemy objects which is extremely challenging, particularly due to complex computational demand for real-time optimal guidance. There was much research reported earlier on how a missile tracks a target and what the optimal guidance law should be used (Zarchan, 1999, 2002; Lin, 1991; Becker, 1990; Hur and Song, 1990; Madkour et al., 2006; Wu et al., 2010; Cui et al., 2011; Shin, 2012; Sun and Xia, 2012). Most of these applications have considered different approaches for adapting the proportional navigation law to improve the performance of the guidance algorithms in two-dimensional environments. Moreover, missile guidance and control algorithms are very time critical. The calculation of the parameters with regard to the missile-target interception requires to be done in a specified time in a 3D defense space. Therefore, missiles guidance can be classified as a hard real-time system. In practice, guidance and control algorithms are very complex and challenging to be implemented in real-time applications.

This investigation proposes a real-time neural network based optimal guidance algorithm for a missile pursuing a maneuvering/nonmaneuvering target in a three-dimensional

* Corresponding author. Tel.: +44 191 243 7089.
E-mail addresses: alamgir.hossain@northumbria.ac.uk (M.A. Hossain), a.madkour@gmail.com (A.A.M. Madkour), k.p.dahal@bradford.ac.uk (K.P. Dahal), li.zhang@northumbria.ac.uk (L. Zhang).

0952-1976/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.engappai.2012.10.007

(3D) environment. In order to achieve a robust guidance system, it needs to address some practical constraints, for instance uncertainty in object identification, natural disturbances etc. This research focuses particularly on the algorithm design in implementing a framework for a missile guidance system by considering that the target tracking system provides necessary parameters. The proposed guidance system considers not only adapting proportional navigation law but also identifying the appropriate time (hence, the position of the target) for launching the missile. This time is referred here as to 'attention time', which is the time from the target's entry to the defense zone to the missile launch. The proposed approach discretises the missile defense (destruction) zone into a large number of patterns. Each pattern is considered as a trajectory identified by 'nine' parameters. An off-line optimizer generates the optimal navigation constant, and the missile attention time for the given pattern based on the target position and its maneuvering parameters. In other words, these two parameters are the outputs for the corresponding set of target input parameters of the optimizer. These generated optimal data are used as training data for the neural network.

In this paper, genetic algorithms (GAs) is used to estimate an optimal set of the effective navigation constant and the missile attention time. The GA is selected as the optimisation technique as it is a parallel global search algorithm and easy to be implemented. It is reported earlier that the Recessive trait Crossover, which is referred to here as (RCGA), offered better performance as compared other GA based optimization approach for problems with a few numbers of variables (Madkour et al., 2007). The same RCGA approach is used in this application. The structure of the neural network used in this application is general regression neural network (GRNN) (Specht, 1991; Kanmani, et al., 2004; Kanbuaa et al., 2005; Hu et al., 2010; Abas, 2011). Finally, the proposed algorithms are implemented, tested and verified through a set of experiments. Their performances are evaluated using a simulation model for a tactical target. A comparative performance of the proposed algorithms is presented and discussed to demonstrate the merits and capabilities of the approach.

2. A proposed guidance framework

The real-time implementation of any optimal guidance algorithm is not easy since a nonlinear two-point boundary value problem needs to be solved to obtain the optimal trajectory. Theoretically, this missile-target dynamic is highly nonlinear due to the fact that the equations of motion are best described in Cartesian coordinate system, whereas aerodynamic forces and moments are conveniently represented in the missile and target body axis system (George and Siouris, 2004). Direct numerical solutions to this problem introduce a heavy in-flight computational burden. The convergence characteristics also may not be guaranteed. Thus solving the problem in real time is often not feasible (Song and Tahk, 2002; Vaidyanathan et al., 2006).

The basic idea of the proposed guidance is to train neural network to learn the functional form of the optimal guidance command in terms of the current states and terminal conditions, and use them for real-time guidance. Fig. 1 shows the block representation of the proposed guidance technique which can be divided into two strides. These include (i) an off-line optimal guidance algorithm to generate training data for the GRNN using the RCGA, and (ii) on-line application of this trained GRNN in real-time missile guidance under the PNG laws.

The off-line optimal algorithm minimizes a performance index which is defined by a combination of the missile flight time f_t, and the miss distance M_d as denoted by

$$j=\min\left(\sqrt{(W_tf_t)^2+(W_dM_d)^2}\right. \quad (1)$$

where W_t and W_d are weighting coefficients.

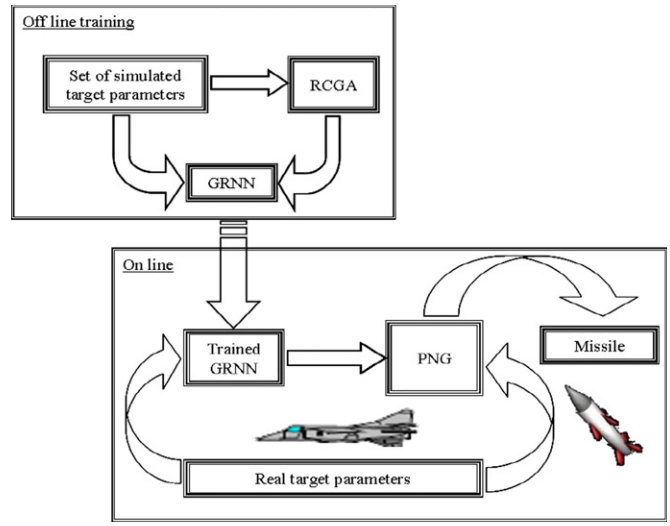

Fig. 1. Neural network guidance technique.

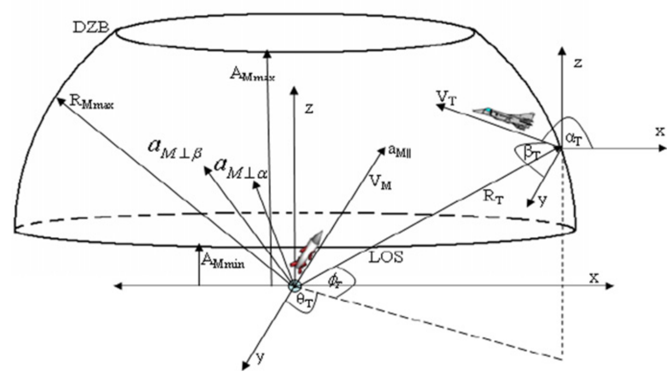

Fig. 2. 3D missile-target engagement geometry and destruction zone boundary (DZB).

This combination gives a trade-off between the two values. In the case of a large variation of these two values, minimizing the sum of squares of the values can be an effective approach (Dahal and Chakpitak, 2007). In fact, there are two parameters which play a key role to minimize this performance index. They are listed below:

i. The effective navigation constant.
ii. The target position at launching time.

It has been reported in the literature that the missile 'flight time' can be greatly reduced by adjusting the effective navigation constant (Lin, 1991; Zarchan, 2002; George and Siouris, 2004; Jackson, 2010). In this application, we consider not only adjusting the navigation constant, but also "optimizing" attention time (hence, the position of the target) for launching the missile.

2.1. Dynamic modelling of the missile and the target

The 3D pursuit geometry of the missile and the target is given in Fig. 2. It is described in a Cartesian coordinate system with origin at the launching point of the missile. The target course can be defined by its velocity vector V_T, and its angles with the vertical and horizontal planes denoted as β_T and $\alpha_{T,}$ respectively. The target measurements are assumed to be obtained from a typical monostatic radar station with origin at the launching

point of the missile with update rate defined by *st*. This measurement set consists of the target range R_T, and its line of sight (LOS) elevation angle θ_T and azimuth angle ϕ_T. The target equation of motion can be written as (Lin et al., 2005),

$$x_T(t) = x_T(t-1) + stV_T(t)\cos\beta(t)\cos\alpha(t) \tag{2}$$

$$y_T(t) = y_T(t-1) + stV_T(t)\cos\beta(t)\sin\alpha(t) \tag{3}$$

$$z_T(t) = z_T(t-1) + stV_T(t)\sin\beta(t) \tag{4}$$

$$V_T(t) = V_T(t-1) + st\,\Delta V(t) \tag{5}$$

$$\beta_T(t) = \beta_T(t-1) + st\,\Delta\beta_T(t) \tag{6}$$

$$\alpha_T(t) = \alpha_T(t-1) + st\,\Delta\alpha_T(t) \tag{7}$$

where $\Delta V(t)$, $\Delta\beta(t)$ and $\Delta\alpha(t)$ are the rate of target velocity vector, angles with the vertical and horizontal planes at scan (t), respectively.

Three acceleration commands are considered for the missile to hit its target at the interception point. Two accelerations are perpendicular to its velocity vector denoted as $a_{M\perp\beta}$ and $a_{M\perp\alpha}$ in the vertical and horizontal planes respectively. The other acceleration is parallel to its velocity vector V_M denoted as $a_{M//}$. These commanded accelerations can be evolved under the pure proportional navigation guidance as (Lin et al., 2005):

$$V_M(t) = V_M(t-1) + st\,a_{M||}(t) \tag{8}$$

$$a_{M//}(t) = \frac{T_M - D_M(t)}{M_M} \tag{9}$$

$$a_{M\perp\beta}(t) = c_N\dot{\theta}_M(t) \tag{10}$$

$$a_{M\perp\alpha}(t) = c_N\dot{\phi}_M(t) \tag{11}$$

where c_N is the proportional navigation constant. The notations $\dot{\theta}_T(t)$ and $\dot{\phi}_M(t)$ represent the line-of-sight (LOS) rate corresponding to the elevation angle and azimuth angle at scan (t), respectively. The notations T_T, D_T and M_T denote the net specific thrust, the drag, and the mass of the missile, respectively. It is worth noting that the turning rate cannot exceed a certain limit. Therefore, to make the proposed model more realistic, we have denoted $a_{M\perp\beta\,\max}$ and $a_{M\perp\alpha\,\max}$ as the maximum missile turning rates of the azimuth and elevation angles respectively. Moreover, we assumed that the missile has maximum range, maximum altitude, and minimum altitude denoted by $R_{M\max}$, $A_{M\max}$, and $A_{M\min}$, respectively; which are shaping its destruction zone boundary (DZB) as shown in Fig. 2. Also, all of these parameters are assumed to be constants.

2.2. DZB and estimation of optimal data using RCGA

To generate the GRNN training data, we discretised the missile destruction zone boundary (DZB) into a large number of patterns with respect to the radar station resolution. Each pattern is a "trajectory" representing missile inputs to strike a target in a certain situation defined by the target position and its maneuvering parameters. The problem of route planning for complex equations of motion may be viewed as one discrete multivariable functional optimization (Vaidyanathan et al., 2001; 2006). It is noted that the target specifications cannot exceed a certain limit, therefore to make the simulation data more realistic, we assumed that the target has maximum velocity vector of $V_{T\max}$, minimum velocity vector of $V_{T\min}$, maximum altitude $A_{T\max}$ and the minimum altitude $A_{T\min}$.The target is assumed to enter the DZB with initial velocity vector denoted by $V_{T(0)}$, and initial range $R_{T(0)}$ at time $t=0$, can be defined as

$$R_{T(t=-st)} > \ R_{M\max} < \ R_{T(t=st)} \tag{12}$$

where

$$R_T(t) = \sqrt{(x_T(t))^2 + (y_T(t))^2 + (z_T(t))^2} \tag{13}$$

and $R_{M\max}$ is the missile maximum range. The pattern can be identified by a set of parameters denoted as Ψ_T (also see Eq. (2)–(7))

$$\Psi_T = [x_T \quad y_T \quad z_T \quad V_T \quad \beta_T \quad \alpha_T \quad \Delta V_T \quad \Delta\beta_T \quad \Delta\alpha_T] \tag{14}$$

This set of parameters will be considered as the GRNN inputs. We assume that the given missile battery does not deal with any target outside its DZB defined by the maximum range $R_{M\max}$, the maximum altitude $A_{M\max}$, the minimum altitude $A_{M\min}$ and the missile maximum flight time (see Fig. 2). The missile maximum flight time is the time taken by the missile from its launching point to the boundary of the destructive zone on its exit if there was no interception with the target. Therefore, not all trajectories are acceptable due to the missile specifications. In fact, any air defense system consists of intersected surface to air missile (SAM) batteries. The acceptable trajectories for a certain missile should achieve a set of conditions defined as below:

i. The initial target position lies at the surface of the destruction zone boundary.
ii. The target trajectory goes into the destruction zone.
iii. The target trajectory altitude is within the destruction zone all the time.
iv. The target remains in the destruction zone not less than the missile maximum flight time $f_{t\max}$.

Solving the target equation of motion (2 to 4) with respect to the time taken to reach the $A_{R_{\max}}$ of the DBZ, and the time taken to reach the maximum and minimum missile altitudes $A_{z_{\max}}$ and $A_{z_{\min}}$, respectively, can be derived as:

$$A_{R_{\max}} = -2\,(z_T\sin(\beta) + x_T\cos(\beta)\,\cos(\alpha) + y_T\cos(\beta)\,\sin(\alpha))/V_T \tag{15}$$

$$A_{z_{\max}} = \frac{z_{\max} - z_T}{V_T\,\sin(\beta)} \tag{16}$$

$$A_{z_{\min}} = \frac{z_{\min} - z_T}{v_T\,\sin(\beta)} \tag{17}$$

The set of conditions for acceptable trajectories can be rewritten as:

$$A_{R_{\max}} > f_{t\max} \quad \text{and} \quad A_{Z_{\max}} < 0 \text{ or } A_{Z_{\max}} > f_{t\max} \quad \text{and} \quad A_{Z_{\min}} < 0 \text{ or } A_{Z_{\min}} > f_{t\max} \tag{18}$$

These conditions are used as search bound for optimization.

The arrangement time delay between the first taking measurements at the missile DZB and the missile fire command (missile launch time) is defined by the *attention time (T_A)*. Then, the multivariable functional optimization problem is summarized by minimizing the performance index given by Eq. (1) interims of the navigation constant (c_N) of the PNG algorithm and the missile attention time (T_A) for the given initial target position. The RCGA is chosen to estimate the interims of the navigation constant (c_N) of the PNG algorithm, and the missile attention time (T_A) for each pattern to minimize the performance index. The performance index is the combination of the missile flight time f_t, and the miss distance M_d denoted by Eq. (1) which assumes to be the RCGA fitness function. The RCGA minimizes this function by tuning the values of the PNG navigation constant (c_N) and the missile attention time (T_A) for every pattern to generate the corresponding output of the neural network. In fact, there is no need to generate the GRNN input data at all around the four Cartesian coordinate quarters. The GRNN can be trained by the first quarter data only as the relationship of the first quarter and

the other three can easily be calculated for the nine target parameters given in Eq. (14) using (19)–(21). It leads to a certain set of proportional navigation constant and attention time. Thus, the equivalent sets in the other three quarters will lead to the same proportional navigation constant and attention time set.

$$[-x_T \quad y_T\, z_T\, v_T \quad \beta_T \quad 180-\alpha_T \quad \Delta V_T \quad \Delta\beta_T \;\; -\Delta\alpha_T] \tag{19}$$

$$[-x_T - y_T\, z_T\, v_T \quad \beta_T\; 180+\alpha_T \quad \Delta V_T \quad \Delta\beta_T\; \Delta\alpha_T] \tag{20}$$

$$[x_T - y_T\, z_T\, v_T \quad \beta_T \quad -\alpha_T \quad \Delta V_T \quad \Delta\beta_T \;\; -\Delta\alpha_T] \tag{21}$$

2.3. Operating procedure to use the algorithm

In this section, a summary is given regarding to how the RCGA–GRNN technique is to be applied for a surface to air missile battery to improve the kill probability of its single shot. This process can be divided into two stages: the off-line stage and the on-line application. The off-line stage is shown as follows:

i. Consider the first quarter of the missile destruction zone boundary to discretise into patterns according to the radar resolution of the surface to air missile battery and its expected target specifications. Each pattern consists of nine parameters given in Eq. (14).
ii. Discard unacceptable sets according to the missile destruction zone altitude boundaries and the target time remaining in this zone with respect to the missile flight time as in Eq. (13).
iii. Estimate the navigation constant and the missile attention time using the RCGA for each acceptable set of target parameters which give minimum missile flight time and miss distance within the missile fuses limitation (Eq. (1)).
iv. Train a GRNN to estimate the navigation constant and the missile attention time for a given target parameters using the accepted target parameters set as an input and the corresponding set of navigation constant and the missile attention time estimated by the RCGA as the output.

The second stage is the on-line application, where the algorithm checks whether the target is valid for DZB according to its nine parameters. Then it estimates the navigation constant and the missile attention for the accepted target using the trained GRNN:

i. Determine the target nine parameters.
ii. Make the decision of accepting or rejecting the target according to the missile destruction zone altitude boundaries and the target time remaining in this zone with respect to the missile flight time (Eq. (13)).
iii. Convert the target nine parameters to their equivalent in the first quarter of the missile destruction zone boundary pattern (Eqs. (14), (19)–(21)).
iv. Estimate the navigation constant and the missile attention time for this first quarter equivalent parameters using the trained GRNN.
v. Wait for this estimated time "missile attention time".
vi. Fire the missile with the estimated navigation constant.

3. Algorithms for optimisation and learning

3.1. RCGA

The Traditional Crossovers Genetic Algorithm (TCGA) works on randomly selected pairs of solutions from mating pool with certain crossover rate. This operation exchanges the genes between the two randomly selected solutions. To reduce the randomization "luck" of the traditional GA crossover operator a modified approach for population inheritance is used referring to the concepts taken from the recessive property inheritance and Darwin theory of evolution. The Recessive trait Crossover Genetic Algorithm (RCGA) produces children by selecting the common genes between both parents, and choosing the remaining genes randomly based on the fact that the complementary of all of the chromosome parts makes its survival fitness. This crossover operator has some similarities with the well known uniform crossover operator.

Table 1
An example of two parent chromosomes having 8 genes.

Gene No.	1	2	3	4	5	6	7	8
Parent 1	0	1	1	1	0	1	0	1
Parent 2	0	0	1	0	0	1	1	0

Table 2
Reproducing of chromosomes given in Table 1 using TCGA.

Gene No.	1	2	3	4	5	6	7	8
Child 1	0	1	1	0	0	1	1	1
Child 2	0	0	1	1	0	1	0	0

Table 3
Reproducing of chromosomes given in Table 1 using RCGA.

Gene No.	1	2	3	4	5	6	7	8
Child 1	0	1	1	1	0	1	0	0
Child 2	0	1	1	0	0	1	1	0
Child 3	0	0	1	0	0	1	0	1
Child 4	0	0	1	1	0	1	1	1

To illuminate the differences between them, we consider two parents have the eight genes chromosome as shown in Table 1. The uniform crossover genetic algorithm can be proceed by exchanging alternate substrings, where the offspring is constructed by choosing every bit with a probability p (usually $p=0.5$ is considered) from either parents, as in Table 2 (Deb, 2001). In the example, the first, fourth, sixth and seventh bits are exchanged between the parents. On the other hand, it has to be noted that these parents have common genes at (1, 3, 5, and 6). Over the evolution process the survival fitness of these two parents depend on these common genes. The RCGA keeps these common genes without any change when the offspring is constructed and tries to make better children by introducing different genes using the four possible binary combinations randomly. This is the only random operation in this crossover approach. The new solutions are shown in Table 3.

Referring to the inheritance of recessive trail properties, the selection of the parents is very important as both of them should have some fitness trait carriers. Further details of RCGA and performance of the algorithm can be found in Madkour et al. (2006).

3.2. GRNN

The general regression neural network (GRNN) was firstly developed by Specht (1991), who claims that the algorithm in GRNN is able to provide a smooth transition from one observed value to another, even with sparse data in a multidimensional measurement space. The GRNN is the NN architecture that can solve any function approximation problem in the sense of

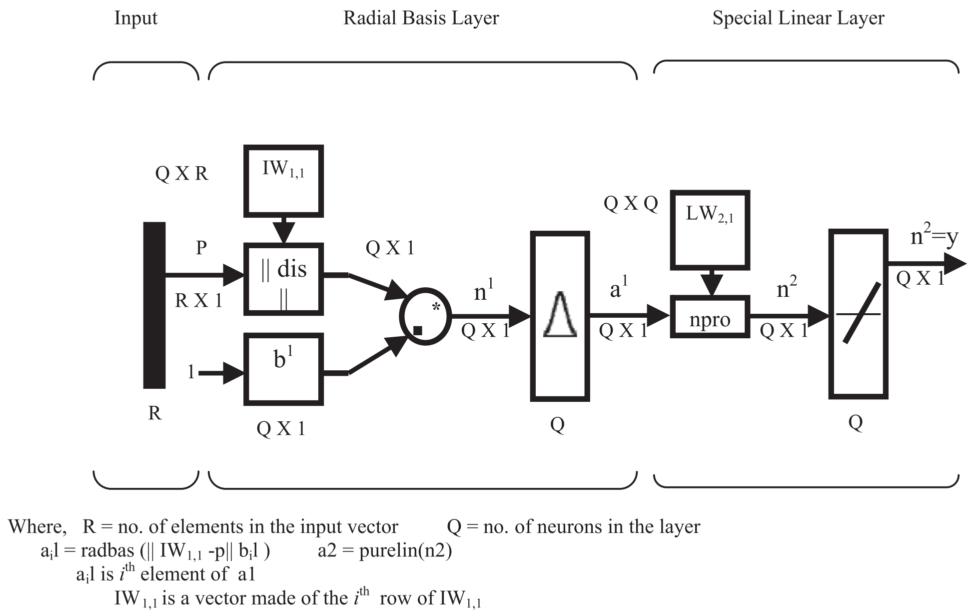

Fig. 3. The basic GRNN structure.

estimating a probability distribution function (Kanbuaa et al., 2005; Hu et al., 2010; Abas, 2011). It is a powerful memory based network that could estimate continuous variables and converge to the underlying regression surface (Kanmani et al., 2004). The main advantage of the GRNN approach is simplicity (Hu et al., 2010). It is noted that the adjustment of only one parameter is sufficient for determining the network. GRNN approximates any arbitrary function between input and output vectors, drawing the function estimate directly from the training data. Furthermore, as the size of the training set becomes larger, the estimation error approaches zero (Kanbuaa et al., 2005).

GRNN is a feed-forward NN established on the theory of non-linear regression. It is a three-layer network with one hidden layer (Kanbuaa et al., 2005; Abas, 2011). Each layer has entirely different roles as shown in Fig. 3.

- *The input layer*: Where the inputs are applied.
- *The hidden layer*: Where a nonlinear transformation is applied on the data from the input space to the hidden space. In most applications the hidden space is of high dimensionality.
- *The linear output layer*: Where the outputs are produced.

4. Experiments and results

A simulation of a maneuvering target is considered to demonstrate the effectiveness of the navigation constant (c_N) and the attention time (T_A) for each pattern by minimizing the performance index denoted in Eq. (1). This target is assumed to enter the missile DZB with initial position, measured in Cartesian coordinate with origin at the launching point of the missile, R_{T0}=[3780,13860,4314] m, with speed $V_{T_0}=220\,m/s$, vertical plane angle $\beta_T=72^\circ$, and horizontal plane angle $\alpha_T=184^\circ$. These values yield the target to be within the missile DZB for about $A_{R_{\max}}=132.87$ s, the time taken to reach the maximum altitude is $A_{z_{\max}}=280.46$ s and a time taken to reach the minimum altitude is $A_{z_{\min}}=-109.86$ s. It is worth noting that the negative value of the time means that it is an imaginary time, which will never be reached. That is, this target will not reach the minimum altitude with these initial parameters. These simulation target trajectories are acceptable according to Eq. (18) as the missile maximum flight time is supposed to be $f_{t\max}=45$ s. This target maneuvers have an acceleration of $\Delta v_T=-3$ m/s^2, vertical plane angle rate of $\Delta\beta_T=2^\circ$/s, and horizontal plane angle rate of $\Delta\alpha_T=4^\circ$/s. The target measurements have been obtained with sampling time st=0.05 s.

The missile flight time f_t, the miss distance M_d, and the performance index denoted by Eq. (1) are calculated to this missile-target pursuit using two sets of navigation constant and attention time denoted by c_N={1, 1.2, 1.4, … ,50}, T_A={1, 1.2, 1.4, …, 50} s, respectively. We have calculated the minimum value of the performance index using exhaustive search for all combinations of c_N and T_A. The minimum value obtained was j=21.0002 which is assumed to be the optimal value for this case. It is worth noting that, the execution time of this experiment was 964.487 s using MATLAB© Version 7.1, which runs on a Dell PC with Intel Pentium 4 CPU 3.2 GHz and 1 GB RAM.

This experiment demonstrates that the navigation constant and the attention time play the key role in determining the flight time and the miss distance (i.e., accurate interception). The good estimation of these parameters (c_N and T_A) leads to a high single-shot kill probability and on the other hand, the bad parameters estimation may lead to a missile failure. In other words, the single-shot kill probability of a guided missile under PNG strongly depends on the estimated values of the navigation constant and the attention time. We propose GAs based optimization to determine the best value of c_N and T_A for every discretised pattern of the DZB. This, however, is a CPU time intensive approach, which cannot be used in real-time. Therefore, GA is used off-line to generate the optimized c_N and T_A. These data are then used to train neural network guidance algorithms for real-time implementation. The next section discusses the application of the GA to generate the training data.

The same specifications of the missile and its maneuvering target discussed above are used for the GRNN model to demonstrate the effectiveness of our optimal guidance algorithm. The minimum value of the performance index obtained using the trained GRNN was j=21.1017. The execution time of this experiment was only 0.025 s. Table 4 summarizes the values of the navigation constant c_N, and the missile attention time T_A given by the TCGA, RCGA and GRNN. It also depicts corresponding miss distance, missile flight time, and the performance index. The last

column indicates the execution time for every case. It is noted that the TCGA performs worst as compared to the other methods. In contrast, the RCGA obtained the solution within 33.078 s as compared to the time taken by the trained GRNN algorithm (0.025 s.). The execution times offered by the TCGA and RCGA algorithms are not suitable for real time implementation. In fact, these algorithms can be used off-line only to generate the GRNN training data to reduce the overall execution time for data generation. The trained GRNN offers best performance in implementing the algorithms within a fraction of a second. Therefore, it clearly demonstrates that the GRNN is the promising algorithm among these for real-time implementation.

Fig. 4 shows the scenario of 3D pursuit trajectory for the simulated missile pursuing its target using PNG for three algorithms. The dotted line in this figure depicts the optimal trajectory, the dashed line depicts the trajectory obtained by the RCGA, and the dash-dot line depicts the trajectory obtained by the GRNN. It is worth noting that, the differences between those trajectories are not easy to be realized in this figure because they are very close to each other.

The same result is shown in Figs. 5–7) with the 2D missile-target pursuit trajectories. The spot represents the launching point while the square represents the target position at the missile DZB.

The change of the target range within the missile DZB is shown in Fig. 8. The spot represents the launching point while the star indicates the interception point. The time before the launching point is the attention time, and the time between the spot and the star is the missile flight time. The path after the star is an imaginary path if there was no interception.

Table 4
Summary of the simulation results.

G. law	c_N	T_A s	M_d cm	f_t s	Execution time (s)
TCGA	26.20	49.90	9.66	21.0	964.487
RCGA	36.17	49.85	1.62	21.0	33.078
GRNN	24.77	49.50	2.70	21.1	0.0250

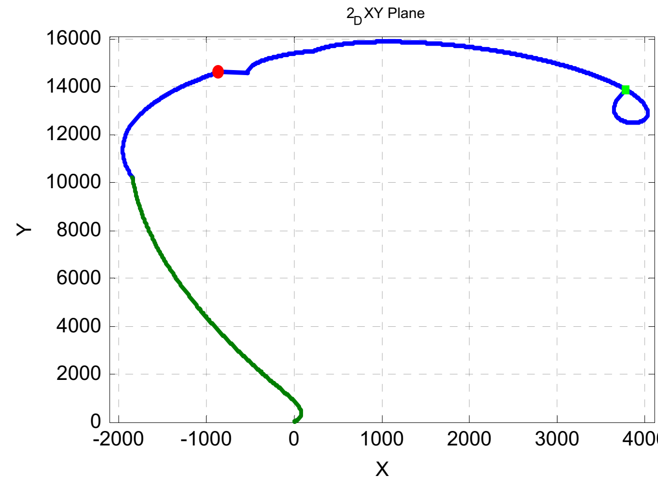

Fig. 5. XY 2D plane pursuit trajectory using the GRNN guidance algorithm (Example 1).

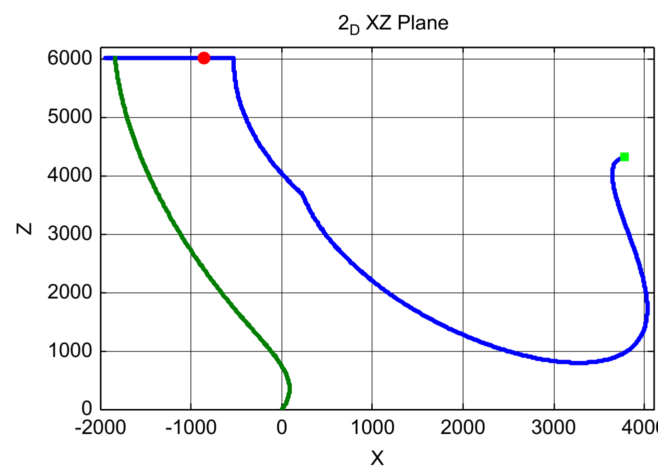

Fig. 6. XZ 2D plane pursuit trajectory using the GRNN guidance algorithm (Example 2).

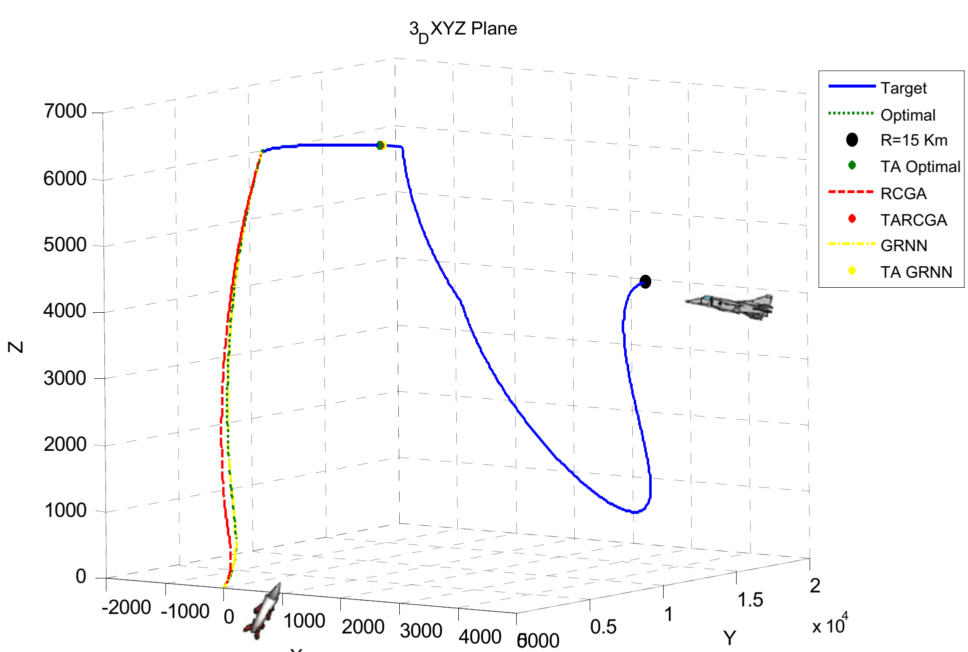

Fig. 4. Three-dimension of pursuit trajectory.

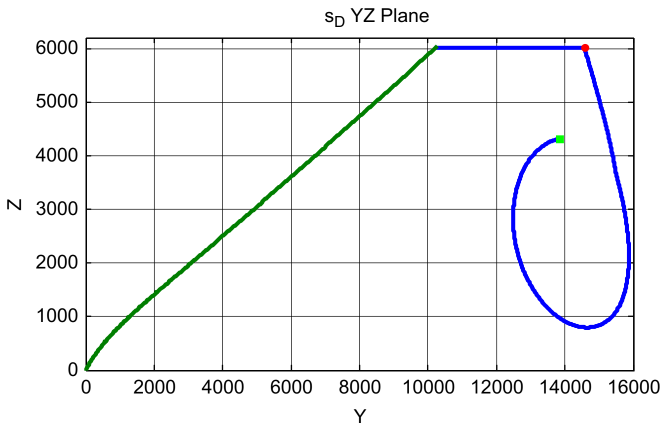

Fig. 7. YZ 2D plane pursuit trajectory using the GRNN guidance algorithm.

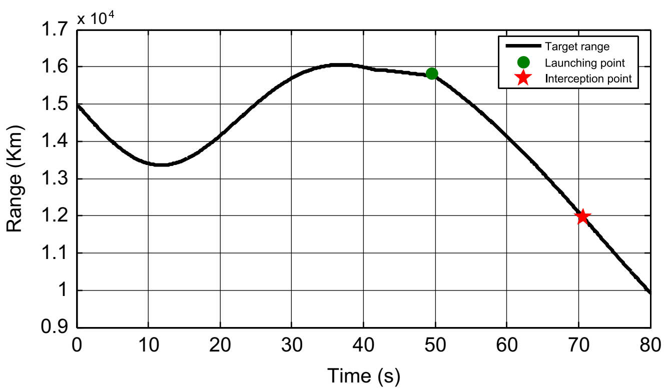

Fig. 8. The simulated target range.

This experiment demonstrates that the proposed GRNN guidance algorithm offers the best performance among the three algorithms in implementing real time system to guide missile to pursue a high maneuvering target to hit it with the minimal flight time and nearly zero miss distance.

5. Concluding remarks

This paper has presented an investigation into the computationally complex optimal real-time missile guidance algorithm to intercept the maneuvering tactical targets in a 3D environment. In this investigation, a new approach of problem discretisation, optimal parameter selection and efficient method to use optimal parameters has been proposed for real-time solution. Here, the input data was generated by discretising the missile defense zone into small patterns. Each pattern was considered as a trajectory identified by 'nine' parameters. The evolutionary genetic algorithm was used as an off-line optimizer to estimate a set of the effective navigation constant and the missile launching time. These parameters have been produced as outputs for the corresponding set of input target parameters. These input and output sets were used to generate a training data set for the GRNN. The trained GRNN was then applied in implementing real-time missile guidance for any unknown target trajectory. The proposed algorithms were implemented, tested and verified through a set of experiments. Their performances were evaluated using a simulation model for a tactical target. A comparative performance of the proposed algorithms is presented and discussed to demonstrate the merits and capabilities of the new approach. We consider a multi-agent based co-operative control approach for multi-missile guidance as our future work.

References

Abas, A.R., 2011. Using incremental general regression neural network for learning mixture models from incomplete data. Egypt. Inf. J. 12 (3), 185–196.

Becker, K., 1990. Closed-form solution of pure proportional navigation. IEEE Trans. Aerosp. Electron. Syst. 26, 526–533.

Benoit, A., Dufossé, F., Girault, A. and Robert, Y. Reliability and performance optimization of pipelined real-time systems. In: The 39th International Conference on Parallel Processing, ICPP 2010, San Diego, CA, USA, 13–16 September 2010. IEEE Computer Society, 2010.

Cui, Y.K., Fu, L., Liang, X.G., Luo, L., 2011. Optimal sliding-mode terminal guidance law design of airborne boost-phase ballistic missile interception. J. Appl. Mech. Mater. 40–41, 15–20.

Dahal, K.P., Chakpitak, N., 2007. Generator maintenance scheduling in power systems using metaheuistic-based hybrid approaches. Electric Power Systems Research 77 (no. 7), 771–779.

Deb, K., 2001. Multi-Objective Optimization using Evolutionary Algorithms. John Wiley & Sons Ltd, pp. 81–169.

George, M., Siouris, 2004. Missile Guidance and Control Systems. Springer Verlag, New York.

Hu, G.N., Liu, C.C., Chuang, K.W., Yu, S.S. and Tsui, T.S. General regression neural network utilized for color transformation between images on RGB color space. In: Proceeding of International Conference on Machine Learning and Cybernetics (ICMLC). China, 2010.

Hur, H.J., Song, T., 1990. Performance analysis of PNG lows for randomly maneuvering targets. IEEE Trans. Aerosp. Electron. Syst. 26, 713–720.

Jackson, P.B., 2010. Overview of missile flight control systems. John Hopkins APL Tech. Dig. 29 (1).

Kanbuaa, W., Supharatidb, S., Tan, I., 2005. Ocean wave forecasting in the Gulf of Thailand during Typhoon Linda 1997: WAM and neural network approaches. J. Sci. Soc. Thailand, Sci. Asia 31, 243–250.

Kanmani, S., Uthariaraj, V.R., Sankaranarayanan, V., Thambidurai, P., 2004. Object oriented software quality prediction using general regression neural networks. ACM SIGSOFT Software Eng. Notes 29, 1–6.

Lin, C.F., 1991. Modern Navigation, Guidance and Control Processing, 2. Prentice Hall.

Lin, L., Kirubarajan, T., Bar-Shalom, Y., 2005. Pursuer identification and time-to-go estimation using passive measurements from an evader. IEEE Trans. Aerosp. Electron. Syst. 41, 190–204.

Madkour, M.A.A., Hossain, M.A. and Dahal, K.P., Evolutionary optimization approach for missile guidance algorithms to achieve accurate interception of maneuvering targets in 3-D. In: Presented at The IEE Seminar on Target Tracking: Algorithms and Applications, UK, 2006.

Madkour, A.A.M., Hossain, M.A., Dahal, K.P., Yu, H., 2007. Intelligent learning algorithms for active vibration control. IEEE Trans. Syst., Man, Cybernet. C: 37 (5), 1022–1033.

Shin, H.S. Study on cooperative missile guidance for area air defence. Ph.D. Thesis. Department of Informatics & Systems Engineering, Cranfield University, UK 2012.

Song, E.J., Tahk, M.J., 2002. Three-dimensional midcourse guidance using neural networks for interception of ballistic targets. IEEE Trans. Aerosp. Electron. Syst. 38, 404–414.

Specht, D.F., 1991. A general regression neural network. IEEE Trans Neural Networks 2, 568–576.

Sun, X., Xia, Y., 2012. Optimal guidance law for cooperative attack of multiple missiles based on optimal control theory. Int. J. Control , http://dx.doi.org/10.1080/00207179.2012.675519.

Vaidyanathan, R., Williams, C., Prince, T., Ritzmann, R., Quinn, R., 2006. An orientation reflex for air vehicles based on a neural model of the cockroach escape response. Global Eng. Sci. Technol. Soc. (GESTS) Int. Trans. Comput. Sci. Eng. 26.

Vaidyanathan, R., Hocaoglu, C., Prince, T. and Quinn, R., Evolutionary path planning for autonomous air vehicles using multi-resolution path representation. In: IEEE International Conference on Intelligent Robots and Systems (IROS), Hawaii, USA and 2001.

Wu, T., Xia, Q. and Du, Y. Optimal guidance law with ability to enlarge range. In: Proceedings of IEEE International Conference on Electronics and Information Engineering (ICEIE), pp. 204–208. 2010. Japan.

Zarchan, P., 1999. Ballistic missile defense guidance and control issues. Sci. Global Security 8, 99–124.

Zarchan, P., 2002. Tactical and Strategic Missile Guidance. U.S.A. AIAA, Inc., Washington, DC, Ed..